VRTRIX Data Glove C++ API Reference

Generated by Doxygen 1.8.16

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

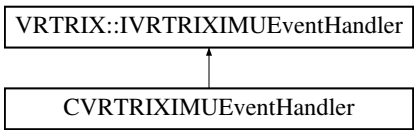
Chapter 3

Class Documentation

3.1 CVRTRIXIMUEventHandler Class Reference

VRTRIX IMU event handler class implementation.

Inheritance diagram for CVRTRIXIMUEventHandler:



Additional Inherited Members

3.1.1 Detailed Description

VRTRIX IMU event handler class implementation.

Implementation of IVRTRIXIMUEventHandler class that handles the IMU event including pose data receiving and other events.

The documentation for this class was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/VRTRIXDataGloveTest/VRTRIXData↩ GloveTest.cpp

3.2 VRTRIX::HandEvent Struct Reference

Glove hand event data structure used in C++ API.

`#include <IVRTRIXIMUEventHandler.h>`

Public Attributes

- HandStatus stat

  Glove hardware status.
- HandType type

  Glove hand type.

3.2.1 Detailed Description

Glove hand event data structure used in C++ API.

The documentation for this struct was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/IVRTRIXIMUEventHandler.h

3.3 VRTRIX::IVRTRIXIMU Class Reference

VRTRIX Data Glove interface class.

`#include <VRTRIX_IMU.h>`

Public Member Functions

- virtual void OpenPort (EIMUError &eError, PortInfo &info)=0
- virtual PortInfo IdentifyPort (EIMUError &eError, HandType type)=0
- virtual void StartDataStreaming (EIMUError &eError, PortInfo info)=0
- virtual void HardwareCalibrate (EIMUError &eError)=0
- virtual void RequestToPair (int channel, PortInfo info, EIMUError &eError)=0
- virtual void RequestToRSSIScan (EIMUError &eError)=0
- virtual void ComPortLatencySetting (PortInfo info, EIMUError &eError)=0
- virtual double GetFingerBendAngle (Joint finger, EIMUError &eError)=0
- virtual double GetFingerYawAngle (Joint origFinger, Joint destFinger, EIMUError &eError)=0
- virtual void Vibrate (EIMUError &eError)=0
- virtual void VibratePeriod (EIMUError &eError, int msDurationMillisec)=0
- virtual void SoftwareAlign (EIMUError &eError)=0
- virtual void SwitchToAdvancedMode (EIMUError &eError, bool advancedMode)=0
- virtual void AlgorithmTuning (EIMUError &eError, Joint finger, AlgorithmConfig type, double value=0, VRTRIXVector_t vec={0, 0, 0})=0
- virtual void RegisterIMUDataCallback (IVRTRIXIMUEventHandler *&pEventHandler, void *pUserParam=NULL)=0
- virtual void ClosePort (EIMUError &eError)=0
- virtual void ConfigHTCTracker (EConfigError &eError, HandType type)=0
- virtual void HandleHTCTrackerEvents (EConfigError &eError)=0
- virtual void IdentifyHTCTracker (EConfigError &eError, HandType type)=0
- virtual int GetWristTrackerIndex (EConfigError &eError, HandType type)=0
- virtual void Uninit ()=0

3.3.1 Detailed Description

VRTRIX Data Glove interface class.

Interface class that contains most APIs for calling

3.3.2 Member Function Documentation

3.3.2.1 AlgorithmTuning()

```
virtual void VRTRIX::IVRTRIXIMU::AlgorithmTuning (
            EIMUError & eError,
            Joint finger,
            AlgorithmConfig type,
            double value = 0,
            VRTRIXVector_t vec = {0, 0, 0} )  [pure virtual]
```

Slerp algorithm tuning

Parameters

eError	return the error during slerp algorithm tuning, if any.
finger	finger joint name (use intermediate joint of each finger to tune algorithm).
type	algorithm configuration type.
value	value to tune.
vec	vector to tune the offset.

Returns

void

3.3.2.2 ClosePort()

```
virtual void VRTRIX::IVRTRIXIMU::ClosePort (
            EIMUError & eError )  [pure virtual]
```

Stop the data streaming and close the serial port

Parameters

eError	return the error during closing the port, if any.

Returns

void

3.3.2.3 ComPortLatencySetting()

```
virtual void VRTRIX::IVRTRIXIMU::ComPortLatencySetting (
            PortInfo info,
            EIMUError & eError )  [pure virtual]
```

Perform COM Port setting Note: This function need to be called when running on a new computer

Parameters

eError	return the error during opening the port, if any.
info	struct contains port information, feed in baud_rate & hand type,

Returns

void.

3.3.2.4 ConfigHTCTracker()

```
virtual void VRTRIX::IVRTRIXIMU::ConfigHTCTracker (
            EConfigError & eError,
            HandType type )  [pure virtual]
```

Config the HTC Tracker with specified handtype

Parameters

eError	return the error during configuration, if any.
type	specify left hand / right hand.

Returns

void

3.3.2.5 GetFingerBendAngle()

```
virtual double VRTRIX::IVRTRIXIMU::GetFingerBendAngle (
            Joint finger,
            EIMUError & eError )  [pure virtual]
```

Get finger bend angle according to specific joint. Note: valid joint index for this function is the intermediate joint for five fingers , which indicates five fingers gesture, any value other than that is NOT allowed and will return error.

Parameters

eError	return the error calculate finger angle, if any.
finger	joint/finger index.

Returns

the specific finger bend angle.

3.3.2.6 GetFingerYawAngle()

```
virtual double VRTRIX::IVRTRIXIMU::GetFingerYawAngle (
            Joint origFinger,
            Joint destFinger,
            EIMUError & eError )  [pure virtual]
```

Get finger yaw angle according to specific joint.

Parameters

eError	return the error calculate finger angle, if any.
origFinger	first joint/finger index.
destFinger	second joint/finger index.

Returns

the specific finger yaw angle between two joints specified.

3.3.2.7 GetWristTrackerIndex()

```
virtual int VRTRIX::IVRTRIXIMU::GetWristTrackerIndex (
            EConfigError & eError,
            HandType type )  [pure virtual]
```

Get Wrist tracker index for left hand & right hand

Parameters

eError	return the error during identify tracker, if any.
type	specify left hand / right hand.

Returns

the index

3.3.2.8 HandleHTCTrackerEvents()

```
virtual void VRTRIX::IVRTRIXIMU::HandleHTCTrackerEvents (
            EConfigError & eError )  [pure virtual]
```

Handle HTC tracker events, should be called in loop;

Parameters

eError	return the error during events handling, if any.

Returns

void

3.3.2.9 HardwareCalibrate()

```
virtual void VRTRIX::IVRTRIXIMU::HardwareCalibrate (
            EIMUError & eError )  [pure virtual]
```

Perform hardware calibrate process. Note: All data gloves have performed IN-FACTORY hardware calibration, no need to do it again
unless the environment magnetic field has been changed dramatically.

Parameters

eError	return the error during calibration, if any.

Returns

void.

3.3.2.10 IdentifyHTCTracker()

```
virtual void VRTRIX::IVRTRIXIMU::IdentifyHTCTracker (
            EConfigError & eError,
            HandType type )  [pure virtual]
```

Identify HTC tracker for left hand & right hand

Parameters

eError	return the error during identify tracker, if any.
type	specify left hand / right hand.

Returns

void

3.3.2.11 IdentifyPort()

```
virtual PortInfo VRTRIX::IVRTRIXIMU::IdentifyPort (
            EIMUError & eError,
            HandType type ) [pure virtual]
```

Identify the corresponding serial port according to handtype.

Parameters

eError	return the error during identifying the port, if any.
type	hand type to identify port.

Returns

port info struct.

3.3.2.12 OpenPort()

```
virtual void VRTRIX::IVRTRIXIMU::OpenPort (
            EIMUError & eError,
            PortInfo & info ) [pure virtual]
```

Open the data streaming serial port according to the PortInfo

Parameters

eError	return the error during opening the port, if any.
info	struct contains port information, feed in baud_rate & hand type, returns other parameters of the port.

Returns

void.

3.3.2.13 RegisterIMUDataCallback()

```
virtual void VRTRIX::IVRTRIXIMU::RegisterIMUDataCallback (
            IVRTRIXIMUEventHandler *& pEventHandler,
            void * pUserParam = NULL )  [pure virtual]
```

Register data call back event handler class.

Parameters

pEventHandler	the call back event handler class.
pUserParam	the user-defined pointer.

Returns

void

3.3.2.14 RequestToPair()

```
virtual void VRTRIX::IVRTRIXIMU::RequestToPair (
            int channel,
            PortInfo info,
            EIMUError & eError )  [pure virtual]
```

Perform hardware pairing Note: All data gloves have performed IN-FACTORY pairing, no need to do it again unless the pairing is lost due to unexpected reason.

Parameters

channel	channel to pair.
info	serial port information.
eError	return the error during pairing, if any.

Returns

void.

3.3.2.15 RequestToRSSIScan()

```
virtual void VRTRIX::IVRTRIXIMU::RequestToRSSIScan (
            EIMUError & eError )  [pure virtual]
```

Reset hardware to rssi noise scan mode

Parameters

eError	return the error during pairing, if any.

Returns

void.

3.3.2.16 SoftwareAlign()

```
virtual void VRTRIX::IVRTRIXIMU::SoftwareAlign (
            EIMUError & eError ) [pure virtual]
```

Align the fingers' orientations to hand.

Parameters

eError	return the error during software alignment, if any.

Returns

void

3.3.2.17 StartDataStreaming()

```
virtual void VRTRIX::IVRTRIXIMU::StartDataStreaming (
            EIMUError & eError,
            PortInfo info ) [pure virtual]
```

Start the data streaming serial port according to the PortInfo Note that this function contains a infinite loop inside, the loop only breaks when error occurs during data streaming.

Parameters

eError	return the error during data streaming, if any.
info	struct contains port information.

Returns

void.

3.3.2.18 SwitchToAdvancedMode()

```
virtual void VRTRIX::IVRTRIXIMU::SwitchToAdvancedMode (
            EIMUError & eError,
            bool advancedMode )  [pure virtual]
```

Switch to advanced mode during runtime

Parameters

eError	return the error during switching to advanced mode during runtime, if any.
advancedMode	advanced mode flag.

Returns

void

3.3.2.19 Uninit()

```
virtual void VRTRIX::IVRTRIXIMU::Uninit ( )  [pure virtual]
```

Uninit the data glove

Returns

void

3.3.2.20 Vibrate()

```
virtual void VRTRIX::IVRTRIXIMU::Vibrate (
            EIMUError & eError )  [pure virtual]
```

Trigger a haptic pulse to the glove.

Parameters

eError	return the error during vibration, if any.

Returns

void.

3.3.2.21 VibratePeriod()

```
virtual void VRTRIX::IVRTRIXIMU::VibratePeriod (
            EIMUError & eError,
            int msDurationMillisec )  [pure virtual]
```

Trigger a period of haptic pulse to the glove.

Parameters

eError	return the error during vibration, if any.
msDurationMillisec	the duration of pulse in millisecond

Returns

void.

The documentation for this class was generated from the following file:

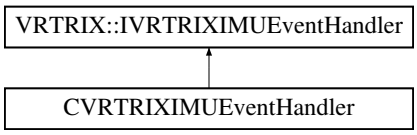
- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/VRTRIX_IMU.h

3.4 VRTRIX::IVRTRIXIMUEventHandler Class Reference

VRTRIX IMU event handler class.

`#include <IVRTRIXIMUEventHandler.h>`

Inheritance diagram for VRTRIX::IVRTRIXIMUEventHandler:



Public Member Functions

- virtual ~IVRTRIXIMUEventHandler (void)
- virtual void OnReceivedNewPose (Pose pose, void *pUserParam)=0
- virtual void OnReceivedNewEvent (HandEvent event, void *pUserParam)=0

3.4.1 Detailed Description

VRTRIX IMU event handler class.

Interface class that define the function header for handling the IMU event including pose data receiving and other events.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 ~IVRTRIXIMUEventHandler()

```
virtual VRTRIX::IVRTRIXIMUEventHandler::~IVRTRIXIMUEventHandler (
            void  )  [inline], [virtual]
```

Uninitialization (Destructor)

3.4.3 Member Function Documentation

3.4.3.1 OnReceivedNewEvent()

```
virtual void VRTRIX::IVRTRIXIMUEventHandler::OnReceivedNewEvent (
            HandEvent event,
            void * pUserParam )  [pure virtual]
```

OnReceivedNewEvent event call back function implement

Parameters

event	Event struct returned by the call back function
pUserParam	user defined parameter

Returns

void

3.4.3.2 OnReceivedNewPose()

```
virtual void VRTRIX::IVRTRIXIMUEventHandler::OnReceivedNewPose (
            Pose pose,
            void * pUserParam )  [pure virtual]
```

OnReceivedNewPose event call back function implement

Parameters

pose	Pose struct returned by the call back function
pUserParam	user defined parameter

Returns

void

The documentation for this class was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/IVRTRIXIMUEventHandler.h

3.5 VRTRIX::PortInfo Struct Reference

Serial port information need for data streaming.

`#include <IVRTRIXIMUEventHandler.h>`

Public Attributes

- std::string port
- std::string description
- std::string hardware_id
- std::string instance_id
- int baud_rate
- HandType type

3.5.1 Detailed Description

Serial port information need for data streaming.

3.5.2 Member Data Documentation

3.5.2.1 baud_rate

`int VRTRIX::PortInfo::baud_rate`

Baud Rate

3.5.2.2 description

`std::string VRTRIX::PortInfo::description`

Human readable description of serial device if available.

3.5.2.3 hardware_id

```
std::string VRTRIX::PortInfo::hardware_id
```

Hardware ID (e.g. VID:PID of USB serial devices) or "n/a" if not available.

3.5.2.4 instance_id

```
std::string VRTRIX::PortInfo::instance_id
```

Instance ID

3.5.2.5 port

```
std::string VRTRIX::PortInfo::port
```

Address of the serial port (this can be passed to the constructor of Serial).

3.5.2.6 type

```
HandType VRTRIX::PortInfo::type
```

Hand Type

The documentation for this struct was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/IVRTRIXIMUEventHandler.h

3.6 VRTRIX::Pose Struct Reference

Glove pose data structure used in C++ API.

```
#include <IVRTRIXIMUEventHandler.h>
```

Public Attributes

- VRTRIXQuaternion_t imuData [Joint_MAX]
  IMU data in quaternion (Global coordinate)
- HandType type
  Glove hand type.
- int calScore [IMU_NUM]
  IMU calibration score. Lower score means better calibration results.
- int radioStrength
  Glove wireless radio strength.
- double battery
  Glove battery percentage.

3.6.1 Detailed Description

Glove pose data structure used in C++ API.

The documentation for this struct was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/IVRTRIXIMUEventHandler.h

3.7 VRTRIX::VRTRIXQuaternion_t Struct Reference

Quaternion data structure used in C++ API.

`#include <IVRTRIXIMUEventHandler.h>`

Public Attributes

- float qx
  x component in quaternion
- float qy
  y component in quaternion
- float qz
  z component in quaternion
- float qw
  w component in quaternion

Friends

- std::ostream & operator<< (std::ostream &o, const VRTRIXQuaternion_t a)
  member operator override

3.7.1 Detailed Description

Quaternion data structure used in C++ API.

The documentation for this struct was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/IVRTRIXIMUEventHandler.h

3.8 VRTRIX::VRTRIXVector_t Struct Reference

3D Vector data structure used in C++ API.

`#include <IVRTRIXIMUEventHandler.h>`

Public Attributes

- float x

  x component in vector
- float y

  y component in vector
- float z

  z component in vector

Friends

- std::ostream & operator<< (std::ostream &o, const VRTRIXVector_t a)

  member operator override

3.8.1 Detailed Description

3D Vector data structure used in C++ API.

The documentation for this struct was generated from the following file:

- D:/VRTRIX/Projects/VRGlove/SDKs/C++/VRTRIXGloveCppSDK/include/IVRTRIXIMUEventHandler.h

Index